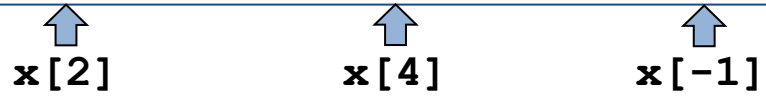
Dict

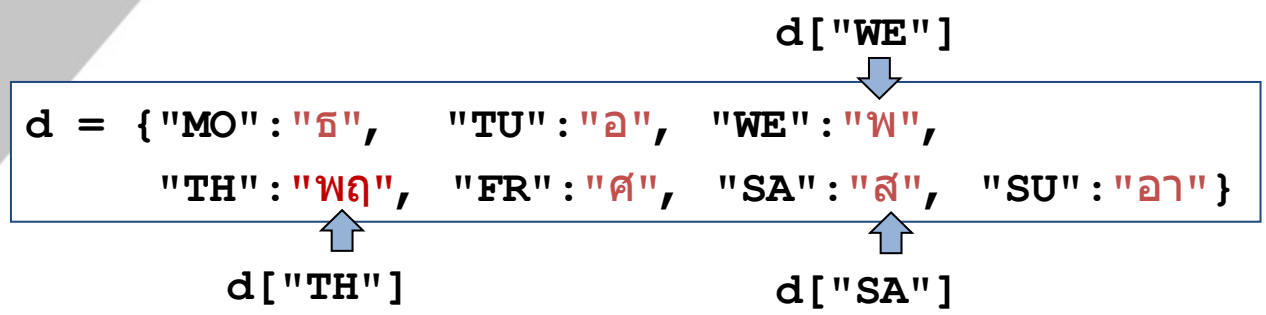
list vs. dict

- list: stores data from left to right
  - use integer as index to access the data in the list

```
x = ["MO", "TU", "WE", "TH", "FR", "SA", "SU"]
```

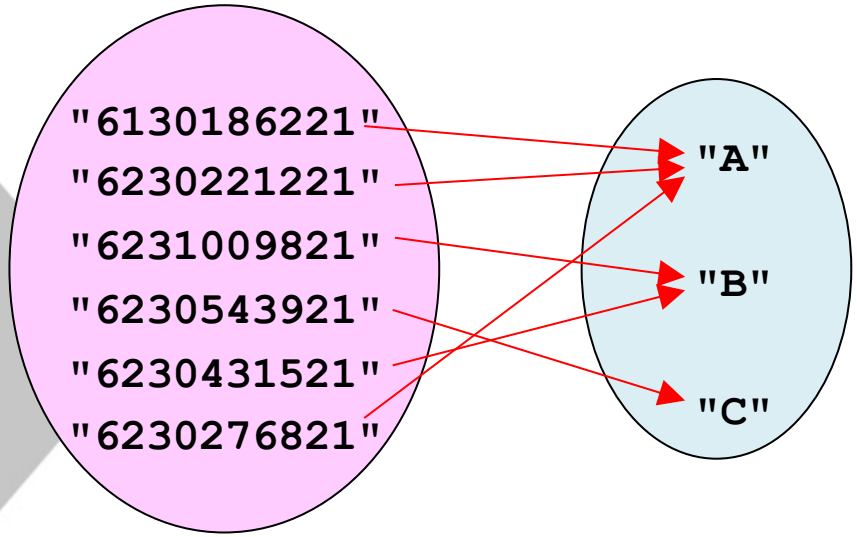


- dict: store data in pairs of *key*: *value*
  - use *key* as index to access the *value* dict
    (*key* can be `int`, `float`, `str`, ...)



dict works like mapping function


```
grade = {"6130186221": "A", "6230221221": "A",
         "6231009821": "B", "6230543921": "C",
         "6230431521": "B", "6230276821": "A"}
```



keys must be unique
values can be duplicated

Specify *key* in dict to get *value*


```
v = grade["6231010621"]
```

key *value*

```
grade = {"6130186221": "A", "6230221221": "A",
         "6231009821": "B", "6230543921": "C",
         "6230431521": "B", "6230276821": "A"}
ID = input()
while ID != "q":
    print(ID, "-->", grade[ ID ])
    ID = input()
```

if the key is not in dict, there will be an error

Input

```
6231009821
6130186221
6230221221
q
```

output

```
6231009821 --> B
6130186221 --> A
6230221221 --> A
```

we can only get from key → value

there is no value → key

Add/update *value* in **dict**


```
grade["6231010621"] =  "A"
```

```
grade = { }
grade["6231010621"] = "A"       # {"6231010621": "A"}

grade["6231009821"] = "B"       # {"6231010621": "A",
                                  "6231009821": "B"}

grade["6231009821"] = "C"       # {"6231010621": "A",
                                  "6231009821": "C"}
```

Add/update *value* in **dict**


```
grade = { } # empty dict
n = int(input())
for k in range(n):
   ID, g = input().split()
   grade[ ID ] = g
print(grade)
```

Input

```
4
6130186221 A
6230221221 A
6231009821 F
6231009821 B
```

add
add
add
update

Output

```
{'6130186221': 'A', '6230221221': 'A', '6231009821': 'B'}
```

Example: Voting

```
BLACKPINK = ["Jisoo", "Jennie", "Rosé", "Lisa"]
votes = {} # create empty dict
for name in BLACKPINK:
  votes[name] = 0 # add new key: value into dict
  # {"Jisoo": 0, "Jennie": 0, "Rosé": 0, "Lisa": 0}

name = input()
while name != 'q':
  if name in BLACKPINK:
    votes[name] += 1 # increment value
  name = input()

for name in BLACKPINK:
  print(name, "-->", votes[name]) # get value
```

```
Lisa
Lisa
Lisa
Lisa
Jennie
Aum
Jisoo
Lisa
q
```

Exercise: alphabet count

```
alphabets = "abcdefghijklmnopqrstuvwxyz"
```

Input

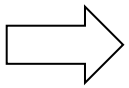
```
ABCabcABZ
```

Output

```
a -> 3
b -> 3
c -> 3
z -> 1
```

for *each_key* in *a_dict*

```
for k in dic:
    if k == t:
        v = dic[k]
```



```
v = dic[t]
```

for *each_key* in *a_dict*

to access each key

```
for k in dic:
    if k == t:
        v = dic[k]
```

```
ordinal = {"first": 1, "second": 2, "third": 3,
           "fourth": 4, "fifth": 5, "sixth": 6,
           "seventh": 7, "eighth": 8,
           "ninth": 9, "tenth": 10 }

for key in ordinal:
    print(key, "-->", ordinal[key] )
```

```
tenth --> 10
second --> 2
fourth --> 4
third --> 3
sixth --> 6
ninth --> 9
seventh --> 7
eighth --> 8
fifth --> 5
first --> 1
```

Note: the order of key from "for" is not the same as the order of creating.

If this program is running with Python 3.7, the order of keys will be the same as the order of creating but it does not guarantee that this behavior will remain in later version of Python. (*Python in Grader also has random order of keys*)

Example: average value of a dic

```
# function average(d)
# d is a dict that have values as number
def average( d ):
    total = 0
    for key in d: # get values from all keys
        total += d[key]
return total / len(d)
```

len(d) is the number of key:value pairs in dict, d

```
gpa = {"6130186221": 3.15, "6230221221": 2.85,
       "6231009821": 2.90, "6230543921": 3.20,
       "6230431521": 3.35, "6230276821": 3.42}

print( average(gpa) ) # should be 3.145
```

Exercise: two functions

```
def reverse( d ): # d is a dict that has unique values
    r = { }
    return r
```

parameter, d `{ "A": "เอ", "B": "บี", "C": "ซี" }`

return `{ "เอ": "A", "บี": "B", "ซี": "C" }`

```
def keys( d, v ): # return list of keys that has value as v
    x = []
    return x
```

parameters `d = { 2:33, 4:55, 9:33, 7:33, 8:55 }`

`v = 33`

return `[ 2, 7, 9 ]`

if *key* in *dict* หรือ if *key* not in *dict*

Is there a *key* in *dict* or not?

```
grade = {"6130186221": "A", "6230221221": "A",
         "6231009821": "B", "6230543921": "C",
         "6230431521": "B", "6230276821": "A"}

ID = input()
while ID != "q":
    if ID in grade:
        print(ID, "-->", grade[ ID ])
    else:
        print("Not found")
    ID = input()
```

if we try to access a key that is not in dict, there will be an error. We must check that the key is in dict before use.

if *key* in *dict* is faster than if *elem* in *list*


```
import time
def search_all(X):
  b = time.time()
  n = len(X)
  for i in range(n):
    if i in X: # True
      pass
  for i in range(n):
    if (n+1) in X: # False
      pass
  print(time.time() - b)
```

```
n = 50000
L = []
for i in range(n):
  L.append(i)
D = {}
for i in range(n):
  D[i] = i

search_all(L)
search_all(D)
```

searching in dict is **much faster** than list

```
40.90408134460449
0.0156097412109375
```

in seconds

Exercise: Nickname, Fullname

Write a program that take nickname and display fullname or takes fullname and display nickname using dict

Robert	⇔	Dick
William	⇔	Bill
James	⇔	Jim
John	⇔	Jack
Margaret	⇔	Peggy
Edward	⇔	Ed
Sarah	⇔	Sally
Andrew	⇔	Andy
Anthony	⇔	Tony
Deborah	⇔	Debbie

http://www.cc.kyoto-su.ac.jp/~trobb/nicklist.html

Example: Vote

```
BLACKPINK = ["Jisoo", "Jennie", "Rosé", "Lisa"]
votes ={"Jisoo":0, "Jennie":0, "Rosé":0, "Lisa":0}

name = input()
while name != 'q':
    if name in BLACKPINK:
        votes[name] += 1
    name = input()

for name in BLACKPINK:
    print(name, "-->", votes[name])
```

```
Lisa
Lisa
Lisa
Lisa
Jennie
Aum
Jisoo
Lisa
q
```

can be checked as key in `votes`

for key in `votes:` is also ok
(next slide)

Example: Vote (II)

```
votes = {"Jisoo":0, "Jennie":0, "Rosé":0, "Lisa":0}

name = input()
while name != 'q':
    if name in votes:
        votes[name] += 1
    name = input()

for name in votes:
    print(name, "-->", votes[name])
```

check that if name is in the votes, before increment the vote

```
Lisa
Lisa
Lisa
Lisa
Jennie
Aum
Jisoo
Lisa
q
```

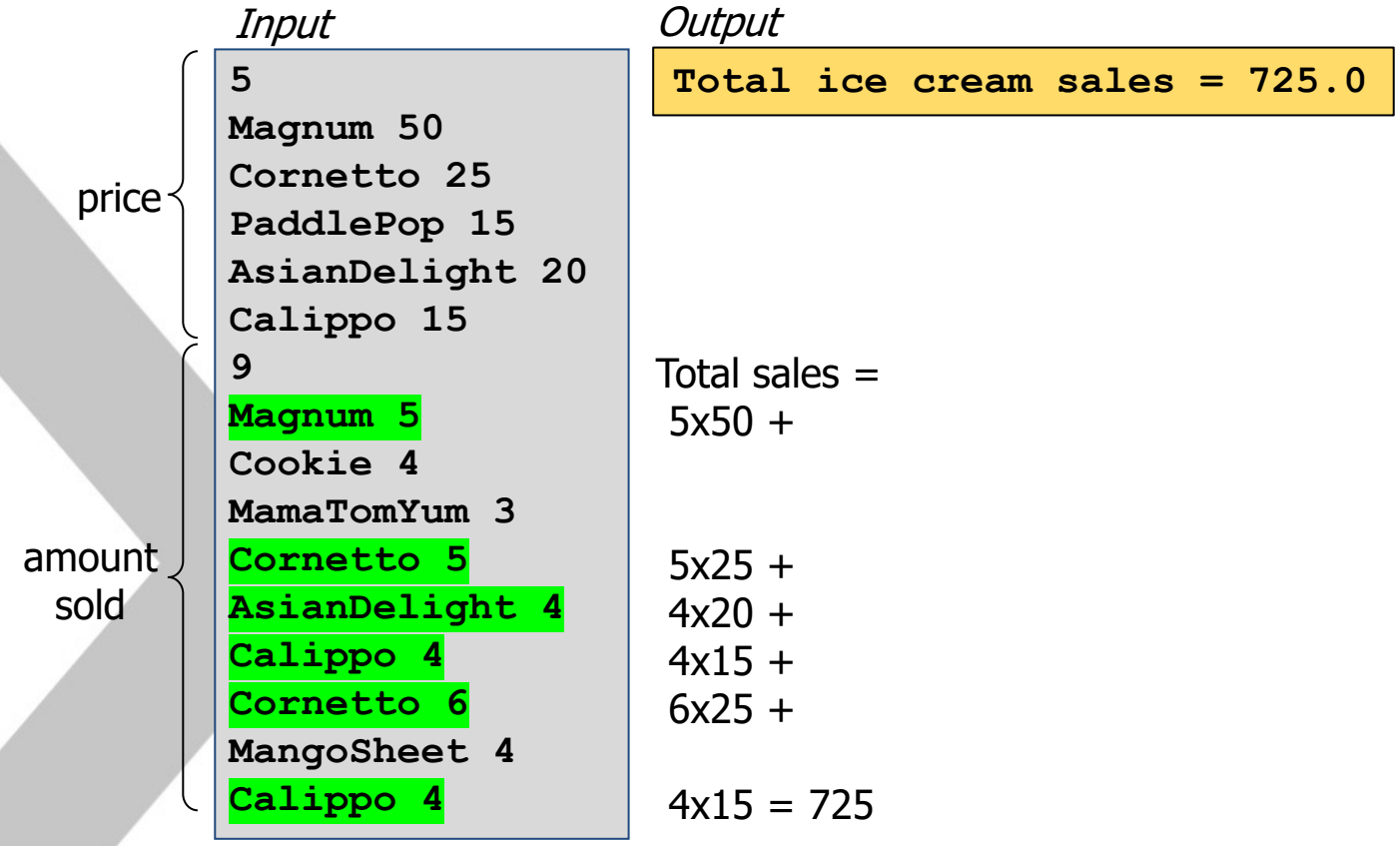
Example: Vote (all names)


```
votes = {}

name = input()
while name != 'q':
    if name in votes:
        votes[name] += 1 # if old name add 1 point
    else:
        votes[name] = 1  # new name give 1 point
    name = input()

for name in votes:
    print(name, "-->", votes[name])
```

We do not know about names. Need to get input, create, then count

```
Lisa
Lisa
Lisa
Lisa
Jennie
Aum
Jisoo
Lisa
q
```

Exercise: Ice Cream Sales



Summary

d is a dict, e is a key

```
if e in d :
    ...
```

fast

x is a list, e is data

```
if e in x :
    ...
```

slow

```
k = x.index(e)
```

slow

k is integer

```
d[e] = z
```

fast

```
x[k] = z
```

fast

```
z = d[e]
```

fast

```
z = x[k]
```

fast